

System for remote loading of objects or files in order to update software

Patent Number: ☐ US6240550
Publication date: 2001-05-29
Inventor(s): DION DOMINIQUE (CA); NATHAN GUY (FR)
Applicant(s): TOUCHTUNES MUSIC CORP (US)
Requested Patent: JP2000099341
Application Number: US19980144440 19980901
Priority Number(s): FR19980009296 19980721
IPC Classification: G06O9/45
EC Classification: G06F9/44G4C
Equivalents: CA2277941, ☐ EP0974896, ☐ FR2781582

Abstract

A system for remote loading of objects or files in order to update software includes operating system architecture that calls for different tasks to be broken down into software modules that are connected to one another by dynamic links or are composed of executable subroutines that have main dependence links to other parts of the operating system. Each of the modules is composed of object files or libraries that are represented by dynamic link libraries among themselves according to a number of dependence levels that are described in their respective attributes

Data supplied from the esp@cenet database - I2

BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-99341

(P2000-99341A)

(43) 公開日 平成12年4月7日(2000.4.7)

(51) Int.Cl.	識別記号	F I	テーマコード(参考)
G 0 6 F 9/445		G 0 6 F 9/08	4 2 0 C
13/00	3 5 1	13/00	3 5 1 H

審査請求 未請求 請求項の数15 O L 外国語出願 (全 32 頁)

(21) 出願番号 特願平11-204535

(22) 出願日 平成11年7月19日(1999.7.19)

(31) 優先権主張番号 9 8 0 9 2 9 6

(32) 優先日 平成10年7月21日(1998.7.21)

(33) 優先権主張国 フランス (F R)

(71) 出願人 599051661

タッチチューンズ・ミュージック・コーポ
レイション

アメリカ合衆国、ネバダ・89104、ラス・
ベガス、イースト・サハラ・1800、スイー
ト・107

(72) 発明者 ギイ・ナタン

フランス国、91330・イエール、リュ・ジ
ヤンヌ・ダルク・1

(74) 代理人 100062007

弁理士 川口 義雄 (外2名)

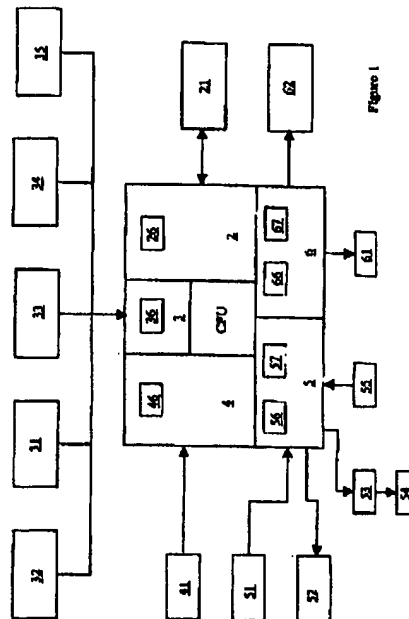
最終頁に続く

(54) 【発明の名称】 ソフトウェアを更新するためにオブジェクトまたはファイルをリモートロードするためのシステム

(57) 【要約】

【課題】 本発明は、ソフトウェアを更新するためにオブジェクトまたはファイルをリモートロードするためのシステムに関する。

【解決手段】 ソフトウェアを更新するためにオブジェクトまたはファイルをリモートロードするためのシステムであって、オペレーティングシステムアーキテクチャは、様々なタスクを、動的リンクによってお互いが接続されたソフトウェアモジュール、あるいは、オペレーティングシステムの他の部分への主従属リンクを有する実行可能なサブルーチンからなるソフトウェアモジュールに分割し、各ソフトウェアモジュールは、動的リンクライブラリによって表現されるオブジェクトファイルまたはライブラリから構成され、動的リンクライブラリは、それらの個々の属性に記述されたいくつかの従属レベルに従ってそれらの間で編成される。



【特許請求の範囲】

【請求項1】 ソフトウェアを更新するためにオブジェクトまたはファイルをリモートロードするためのシステムであって、

オペレーティングシステムアーキテクチャは、様々なタスクを、動的リンクによってお互いが接続されたソフトウェアモジュール、あるいは、オペレーティングシステムの他の部分への主従属リンクを有する実行可能なサブルーチンからなるソフトウェアモジュールに分割し、各ソフトウェアモジュールは、動的リンクライブラリによって表現されるオブジェクトファイルまたはライブラリから構成され、動的リンクライブラリは、それらの個々の属性に記述されたいくつかの従属レベルに従ってそれらの間で編成されるシステム。

【請求項2】 オブジェクトまたはライブラリの属性が、バージョン番号と、他のオブジェクトに対するそのオブジェクトの従属関係とを示す請求項1に記載のシステム。

【請求項3】 属性が、ソフトウェアモジュールが属するレベルを示す請求項2に記載のシステム。

【請求項4】 様々なタスクが、モジュール(JUKECORE, DLL)を含むメインタスクを含み、該モジュール(JUKECORE, DLL)は、動的リンクライブラリ(DLL)をロードし、プログラムのカーネルを初期化し、グラフィック管理モジュール(GFX, DLL)を初期化し、ライブラリロードモジュール(WDLL, DLL)を初期化し、通信タスクであるテレコムモジュール(TELECOM, DLL)をロードし、メインタスクとしてスクリーンインタプリタプログラムを起動することを目的とする、請求項1から3のいずれか項に記載のシステム。

【請求項5】 プログラムがいくつかのモジュールに再分割され、そのそれぞれが、少なくとも通信モデムおよび中央処理装置を備えた端末に固有のタスクを定義する請求項4に記載のシステム。

【請求項6】 前記固有のタスクが、固有のメインタスクを含むジュークボックスに対応したタスクであり、さらに、ディスプレイのウィンドウ機能を規定するいくつかのモジュールを含み、該モジュールには、

- マウスあるいはタッチスクリーン信号を利用するためのMOUSE, DLLモジュール、
- オブジェクト間で交換されるメッセージを利用するためのWMESSEGE, DLLモジュール、
- ディスク上のファイルを管理するためのFILE, DLLモジュール、
- ディスク上のファイルを読み出し書き込むためのFILEIO, DLLモジュール、

タッチスクリーン、サウンドカード、および、硬貨装置のための処理インタフェースのようなハードウェアによって生成されるすべてのイベントを監視するための

JSTRUCT, DLLモジュール、

が含まれる請求項5に記載のシステム。

【請求項7】 ジュークボックスアプリケーションの固有のメインタスクが、ローダプログラムのライブラリを含むモジュール(SILOAD, DLL)を含み、ローダプログラムのライブラリは、必要とされる動的リンクライブラリのバージョンを確認し、それらをロードし、かつ、ファイルを転送するためにテレコムタスクモジュールを呼び出すように設計された請求項5または6に記載のシステム。

【請求項8】 前記SILOAD, DLLモジュールが、動作に必要な最低限のバージョンに関するリストと、ライブラリ(DATAJUKEBOXLIBRARY)によってかまたはオブジェクトファイル(DJO, DATAJUKEBOX OBJECT)によって表現されるすべての機能に関するリストとをファイル(DLL, DEFAULT)に含む請求項7に記載のシステム。

【請求項9】 それぞれのオブジェクトまたはライブラリが、オブジェクトまたはライブラリが必要とするすべての機能に関するリストと、バージョン番号および従属関係とを含む請求項8に記載のシステム。

【請求項10】 WDLL, DLLモジュールが、新しいモジュールを管理し、かつ、リモートロードされたモジュールが従属関係を失っていないこと、およびそれらが必要とされるバージョンに一致してロードされたことを確認するための手段を含む請求項4に記載のシステム。

【請求項11】 SILOAD, DLLモジュールが、端末のタスクに固有なモジュールのロードを管理するための手段を含み、該固有なモジュールには、すべてのDLLモジュール、オブジェクト、ミキサー、および、購入を管理するWOBJECT, DLLモジュールから構成されるジュークボックスライブラリモジュール、カーソルの移動を管理するWCURSOR, DLLモジュール、データベースを管理するDBMAPI, DLLモジュール、フォントタイプを管理するWFFONTS, DLLモジュール、スクリプトに基づいてスクリーンを解析および生成し、かつ、"GRAMMAR, DLL"モジュール及びレキシカルモジュールである"LEXY, DLL"モジュールの支援によって文法を確認するPARSER, DLLモジュールが含まれる請求項7に記載のシステム。

【請求項12】 SILOAD, DLLライブラリロードモジュールが、"WINDEF"モジュールを含み、該"WINDEF"モジュールが、ジュークボックス型端末のモニター上に表示されるディスプレイのウィンドウを管理するために含まれなければならないファイルのリストを含む請求項11に記載のシステム。

【請求項13】 前記オブジェクトのリストが、

10

20

30

40

50

ー モニター上のメインページを規定するのを可能にする“OBJECT WPSCREEN, DJO”オブジェクトモジュール、

ー このメインページで利用可能なスクリーンの数を規定するのを可能にし、それによって、複数のウィンドウまたはスクリーンを表示するのを可能にする“WSCREEN, DJO”モジュール、

ー スクリーンが使用する画像をスクリーン上に決定および規定するのを可能にする“WIMAGE, DJO”モジュール、

ー 画像が動画化される場合の動画を規定するのを可能にする“WANIM, DJO”モジュール、

ー メインページのスクリーン上で使用されるボタンを規定および管理するのを可能にする“WBUTTON, DJO”モジュール、

ー オブジェクトに書き込むのを可能にするラベルを生成するのを可能にする“WLABEL, DJO”モジュール、

ー 2点間でスクロールされる表示領域を規定するのを可能にする“WSCROLLER, DJO”モジュール、

から構成される請求項12に記載のシステム。

【請求項14】 メインタスクによって管理されるこれらのオブジェクトモジュールのすべてが、“JHANDLER”ライブラリを使用し、該“JHANDLER”ライブラリは、スクリーンの所定の使用法を規定するのを可能にし、したがって、上述のソフトウェアモジュールによって規定された様々なオブジェクトへのリンクを保証するのはどのインタフェースかを決定するのを可能にする請求項13に記載のシステム。

【請求項15】 SLOAD, DLLタスクが、“XCP, DJL”モジュールを起動またはロードするための手段を含み、該“XCP, DJL”モジュールは、チケット受領システムあるいは硬貨またはカードによる支払いユニットのような支払いタスクを管理するのを可能にし、また、IBUTTON内で基本情報を保護するのを可能にする請求項7に記載のシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ソフトウェアを更新するためにオブジェクトまたはファイルをリモートロードするためのシステムに関し、より詳細には、ジュークボックスあるいはその他の装置のような料金を支払うことで始動するオーディオビジュアル再生システムに関する。

【0002】

【従来の技術】従来の技術に関しては、例えば、英国特許第2,231,180号などから、ネットワークを介してオペレーティングシステムをリモートロードするための装置が既知である。この特許出願の要件は、オペレ

ーティングシステムのサブセットを第1のコンピュータのメモリにロードすることによって、ネットワークを介して第2のコンピュータに接続された第1のコンピュータにロードすることを要求するものであり、前記サブセットは、ファイルをコピーし、ディレクトリを作成し、ディスクをフォーマットすることを可能にするコマンドを含み、さらに、ネットワークを介してのコネクションを動作させ、それによって、その後、このサブセットは、第2のコンピュータから第1のコンピュータのディスクにオペレーティングシステムファイルのすべてを転送するのに使用することができる。

【0003】この種のリモートロードは、オペレーティングシステム全体すなわちすべてのオペレーティングシステムファイルをロードすることを目的とするものである。したがって、これは、オペレーティングシステムの場合には、リモートロードを実行すべき通信システムが比較的に長い時間にわたりリモートロードによって占有されるという、あるいはその代わりに、関連するファイルを頻繁に更新しなければならないという制限がある。

【0004】米国特許第4,958,278号からは、ディスクプレイヤーを備えていない端末にリモートロードするためのシステムを知ることができる。

【0005】フランス特許第2,682,786号からは、ハードディスクを備えていない端末にリモートロードするためのさらなる方法を知ることができる。

【0006】最後に、ヨーロッパ特許第0,358,992号は、オペレーティングシステムまたは実行可能ファイルを、ネットワークを介して、ブート装置または実行可能プログラムを記録する別の装置を含まないコンピュータにリモートロードするための方法を開示する。必要最小限の第1のブートプログラムが、まず最初に転送され、そして、この最低限のブートプログラムが、それ自身を実行してサーバのディスクとの論理リンクを確立し、それによって、要求コンピュータは、サーバディスクをローカルブート装置とみなすことができる。

【0007】

【発明が解決しようとする課題】本発明の目的は、一方において、ダウンロードがなされる端末をリポートしなくてもよくすることであり、また他方においては、装置の再初期化を必要とせず、オペレーティングファイルまたは実行可能プログラムの一部を転送するのを可能にすることであり、かつ、リモートロードされたバージョンによってシステムの動作をまったく低下させることなくそれを実行することである。

【0008】

【課題を解決するための手段】この目的は、オペレーティングシステムのアーキテクチャが、様々なタスクをソフトウェアモジュールに分割することによって達成され、このソフトウェアモジュールは、動的なリンクによってお互いに接続され、あるいは、オペレーティングシ

システムの他の部分への主従属リンクを有する実行可能なサブルーチンから構成され、それによって、それぞれのモジュールは、動的リンクライブラリによって表現されるオブジェクトファイルまたはライブラリから構成され、動的リンクライブラリは、それらの個々の属性に記述されたいくつかの従属レベルに従ってそれらの間でお互いに編成される。

【0009】本発明のさらなる特徴によれば、オブジェクトまたはライブラリの属性は、バージョン番号と、他のオブジェクトに対してのそのオブジェクトの従属関係とを示す。

【0010】本発明のさらなる特徴によれば、属性は、モジュールが属するレベルを示す。

【0011】本発明のさらなる特徴によれば、様々なタスクは、モジュール(JUKECORE)を含むメインタスクを含み、モジュール(JUKECORE)は、動的リンクライブラリ(DLL)をロードし、プログラムの核を初期化し、グラフィック管理モジュール(GFX)を初期化し、ライブラリロードモジュール(WDLL)を初期化し、通信タスクであるテレコムモジュール(TELECOM, DLL)をロードし、メインタスクとしてスクリーンインタプリタプログラムを起動することを目的とする。

【0012】本発明のさらなる特徴によれば、プログラムは、端末に固有のタスクを定義するいくつかのモジュールに再分割される。

【0013】本発明のさらなる特徴によれば、この固有のタスクは、固有のメインタスクを含むジュークボックスに対応したタスクであり、さらに、ディスプレイの“ウィンドウズ”機能を規定するいくつかのモジュールを含み、該モジュールには、

マウスあるいはタッチスクリーン信号を処理するモジュール、

— オブジェクト間で交換されるメッセージを処理するためのモジュール、

— ディスク上のファイルを管理するための“FIL, DJL”モジュール、

— ディスク上のファイルを読み出したディスクにファイルを書き込むための“FIL, DJL”モジュール、

— タッチスクリーン、サウンドカード、および、マネー用ハードウェア処理インタフェースのようなハードウェアによって生成されるすべてのイベントを監視するための“FILLIO, DJL”モジュールが含まれる。

【0014】本発明のさらなる特徴によれば、ジュークボックスアプリケーションのメインタスクは、ロードプログラムのライブラリを含む“SILLOAD, DLL”モジュールを含み、それによってこのロードプログラムのライブラリは、必要とされる動的リンクライブラリの

バージョンを確認し、それらをロードし、かつ、ファイルを転送するためにテレコムタスクモジュールを呼び出すためのものである。

【0015】本発明のさらなる特徴によれば、SILOADモジュールは、動作に必要な最低限のバージョンに関するリストと、ライブラリ(DJL)(DATA JUKEBOX LIBRARY)によってかまたはオブジェクトファイル(DJODATA JUKEBOX OBJECT)によって表現されるすべての機能に関するリストとをファイル(DLL, DEFAULT)に含む。

【0016】本発明のさらなる特徴によれば、オブジェクトまたはライブラリは、オブジェクトまたはライブラリが必要とするすべての機能に関するリストと、バージョン番号および従属関係とを含む。

【0017】本発明のさらなる特徴によれば、WDLLモジュールは、すべての新しいモジュールの管理を保証し、かつ、リモートロードされたモジュールがいかなる従属関係も失っていないこと、およびそれらが必要とされるバージョンに一致してロードされたことを確認する。

【0018】本発明のさらなる特徴によれば、SILOADモジュールは、端末のタスクに固有なモジュール、すなわち、上に一覧表として記載したすべての“DJL”モジュール、さらに、オブジェクト、ミキサー、および、購入を管理するWOBJECTから構成されるジュークボックスライブラリモジュール、カーソルの移動を管理する“WCURSOR”モジュール、データベースを管理するDBMAPモジュール、すべてのフォントタイプを管理する“WFFONTS”モジュール、スク립トに基づいてスクリーンを解析および生成し、かつ、“GRAMMAR, DJL”モジュール及びレキシカルモジュールである“LEXY, DJL”モジュールの支援によって文法を確認する“PARSER”モジュール、のロードを管理する。

【0019】本発明のさらなる特徴によれば、ライブラリロードモジュールSILLOADは、“WINDEF”モジュールを含み、この“WINDEF”モジュールは、ジュークボックス型端末のモニタ上に表示されるウィンドウズディスプレイのウィンドウを管理するために含まれなければならないファイルのリストを含む。

【0020】本発明のさらなる特徴によれば、このオブジェクトのリストは、

— モニタ上のメインページを規定するのを可能にする“OBJECT WPSCREEN, DJO”モジュール、

このメインページで利用可能なスクリーンの数を規定するのを可能にし、それによって、複数のウィンドウまたはスクリーンを表示するのを可能にする“WSCREEN, DJO”オブジェクトモジュール、

- ー スクリーンが使用する画像をスクリーン上に決定および規定するのを可能にする“WIMAGE、DJO”モジュール、
- ー 画像が動画化される場合の動画を規定するのを可能にする“WANIM、DJO”モジュール、
- ー メインページのスクリーン上で使用されるボタンを規定および管理するのを可能にする“WBUTTON、DJO”モジュール、
- ー オブジェクトの先頭に書き込むのを可能にするラベルを生成するのを可能にする“WLABEL、DJO”モジュール、

2点間で例えば水平、対角線、垂直なスクロール表示領域を規定するのを可能にする“WSCROLLER、DJO”モジュール、から構成される。

【0021】本発明のさらなる特徴によれば、メインタスクによって管理されるこれらのオブジェクトモジュールのすべては、“JHANDLER”ライブラリを使用し、“JHANDLER”ライブラリは、スクリーンの所定の使用法を規定するのを可能にし、したがって、上述のソフトウェアモジュールによって規定された様々なオブジェクトへのリンクを保証するのはどのインタフェースかを決定するのを可能にする。

【0022】本発明のさらなる特徴によれば、SILO ADタスクは、“XCP”モジュールを起動またはロードし、チケット受領システムあるいは硬貨またはカードによる支払いユニットによって処理されるような支払いタスクを管理するのを可能にし、また、IBUTTONによって基本情報を保存するのを可能にする。

【0023】添付の図面を参照して以下の説明を理解することによって、本発明のその他の特徴および利点がより明確となるであろう。

【0024】

【発明の実施の形態】オーディオビジュアル再生システムは、それに限定しないが好ましくは、以下に一覧表として記載したハードウェア構成要素を使用する。

【0025】図1を参照すると、マイクロプロセッサ中央処理装置1は、高性能PC互換システムであり、実施例においては、以下のメモリ資源および仕様を有するPentium型システムが選択された。すなわち、

- ー VESAローカルバスとの互換性、
- ー プロセッサキャッシュ：最低限256KB、
- ー RAMメモリ：32MB、
- ー 高性能シリアルポート、
- ー SVGAマイクロプロセッサグラフィックアダプタ、

SCSI/2型バスコントローラ、

- ー 自己電源スタティックRAMメモリ。

【0026】同等かまたはそれ以上の性能を有していれば、本発明においては、どのような中央処理装置が使用

されてもかまわない。

【0027】この中央処理装置は、音声制御回路5、通信制御回路4、入力制御回路3、大容量記憶装置制御回路2、および、ディスプレイ制御回路6を制御および管理する。ディスプレイは、基本的には、フラットスクリーンノンインタレース方式SVGA高解像度低放熱ビデオモニター62から構成され、これは、画像（例えば、逆曲されたアルバムカバー）、グラフィックス、または、ビデオクリップを再生するのに使用されるモニターである。

【0028】高速かつ大容量のSCSI型ハードディスクを使用した大容量記憶装置21が、マイクロプロセッサ装置にすでに存在している記憶装置に結合される。これらの記憶装置は、デジタル圧縮オーディオビジュアル情報を記憶するのに使用できる。

【0029】高速通信モデムアダプタ1が組み込まれて、中央サーバによって制御されるオーディオビジュアル情報配信ネットワークへのリンクを可能にする。選曲された音声情報を再生するために、システムは、同調増幅器53からの信号を受信する拡声器54を含み、この同調増幅器53は、きわめて多くの入力源をサポートするために提供されたミュージックシンセサイザー型の2つのメモリバッファ58および57を備えた電子回路5に接続され、CD（コンパクトディスク）型の出力品質を提供する。

【0030】同様に、ディスプレイ制御回路6も上述の目的のために2つのバッファメモリ68および67を有する。

【0031】240ワットの温度調整換気電源が、システムに電力を供給する。この電源は、過電流および過熱から保護される。

【0032】オーディオビジュアル再生システムは、その入力コントローラ回路3によって、ATタイプバスコントローラとともに、“最先端表面波技術”を用いたガラスコーティングパネルを含むタッチスクリーン33を管理する。このタッチスクリーン33は、ビデオモニター62かまたはテレビ受像機61のスクリーン上に、システムの管理者または所有者が使用する管理コマンドおよび制御データだけでなく、顧客が使用する様々な選択データも表示することができる。さらに、このタッチスクリーンは、システムに接続されてもよい外部キーボード34と組み合わせて、保守のためにも使用され、システムは、そのために、インタフェース回路3を介してキーロック32によって制御されるキーボードコネクタを有する。

【0033】また、入力回路3は、例えば無線周波数RDRリモコンからなる遠隔制御アセンブリ31とシステムとのインタフェースをとる。

【0034】さらに、料金支払い装置35が、入力インタフェース回路3に接続される。硬貨、チケット、ト

クン、磁気カードまたはマイクロチップカード、または、これらの支払い手段のあらゆる組み合わせを用いたどのような種類の支払いでも受け入れることのできる他のいかなる装置を使用してもよい。

【0035】システムを設置できるように、それは、シャーシを備えるか、または、特注可能な外部付属品を備えて構成される。

【0036】これらの構成要素に加えて、ワイヤレスマイクロホン55が音声コントローラ5に接続され、これは、音声コントローラを強力な放送用情報システムかあるいはオプションとしてカラオケ装置に変える。同様に、システムは、ワイヤレス拡声器システムを使用することもできる。

【0037】遠隔制御アセンブリ31は、管理者が例えばカウンタ越しにアクセスして以下のような様々なコマンドを制御するのを可能にする。すなわち、

マイクロホンのオン/オフ制御、

拡声器のミュート制御、

— 現在演奏されている選曲をキャンセルするための制御。

【0038】2つのバッファ56および57は、少なくとも約1/4秒の音声に相当するデータアイテムをそれぞれが交互に記憶するのを可能にするために、音声コントローラ回路5に結合される。同様に、2つのバッファ66および67は、ビデオコントローラ回路6に結合され、それによって、それぞれのバッファは、それ自身によってかまたは交互に少なくとも約1/10秒の画像を記憶することができる。最後に、各バッファ46、36、および、26は、それぞれ、通信コントローラ回路4、入力インタフェース回路3、および、記憶回路2に結合される。

【0039】システムオペレーティングソフトウェアは、マルチメディア分野のオーディオビジュアル領域をきわめて強く指向したツールおよびサービスのライブラリを中心にして開発されたものである。このライブラリは、有益なことに、複数のコードフラグメントを同時に実行するのを効率的に可能にする高性能マルチタスクオペレーティングシステムを含む。このオペレーティングシステムは、また、表示および音声を再生する構造で実行される演算を並行して順番におよび全く矛盾のない方法で実行するのを可能にするだけでなく、配信ネットワークを介しての通信リンクを管理することも可能にする。さらに、このソフトウェアは高い柔軟性を有する。

【0040】オペレーティングシステムは、第1のブートモジュール7を含む複数のモジュールに分割され、第1のブートモジュール7は、メモリを検査しかつ必要最小限の数のオブジェクトが利用できるかどうかを確認してジョークボックスの動作を保証する“JUK. EX”と呼ばれる第1のメインプログラムモジュール70にさらに分割され、それに動的にリンクし従属する第2

のモジュール71は、“JUKECORE. DLL”と呼ばれるモジュールで表される。第2のモジュール71の機能は、Cによるライブラリを保持することであり、そして、メインタスクを確実に実行させることである。

【0041】オペレーティングシステムのアーキテクチャは、様々なタスクが、動的なリンクによってお互いが接続されたソフトウェアモジュール、あるいは、オペレーティングシステムの他の部分への主従属リンクを有する実行可能なサブルーチンからなるソフトウェアモジュールに分割されることを必要とする。各モジュールは、属性に記述されたいくつかの従属レベルに従って編成されたオブジェクトファイルまたは動的なリンクライブラリファイルから構成される。オブジェクトまたはライブラリの属性は、以下で“PARSER”と呼ばれるモジュールについて説明するように、バージョン番号と、その他のオブジェクトファイルに対してのそのオブジェクトファイルまたはライブラリファイルの従属関係を示す。各属性は、モジュールのレベル属性を示す。したがって、JUK. EXEモジュール70は、JUKECORE. DLL71、TLS. DLL72、CRDE. DLL73、GFX. DLL74、WDDL. DLL75、JEEP. DLL9、および、TELECOM. DLL10のモジュールよりも高いレベルを有するが、TELECOM. DLLモジュール10は、JEEP. DLLモジュール9に従属するので（リンク910を参照）、JEEP. DLLモジュール9よりもレベルが低い。

【0042】同様に、JEEP. DLLモジュール9（リンク759を参照）は、WDDL. DLLモジュール75よりも低いレベルを有する。なぜなら、JEEP. DLLモジュール9は、WDDL. DLLモジュール75に従属するからである。同様に、TLS. DLLモジュール72は、WDDL. DLLモジュール75よりも高いレベルを有する。しかしながら、TLS. DLLモジュールとGFX. DLLモジュールとは、同一レベルであってもよい。メインタスクは、モジュール（JUKECORE. DLL）を含み、その目的は、“CRDE. DLL”と呼ばれるプログラムの核であるモジュール73を初期化またはロードし、グラフィックスを管理するためのモジュール74（GFX. DLL）を初期化またはロードし、ライブラリをロードするためのモジュール75（WDDL. DLL）を初期化またはロードし、通信タスク（DLL）であるテレコムモジュール10をロードし、ジョークボックスに必要な、通信、時刻、解読などのすべてに使用されるTLS. DLLモジュール72をロードし、インテグリティサーバのタスク、ロード要求、および、サーバとの対話を処理するJEEP. DLLプログラム（Juke Execution Exchange Protocol（ジョーク実行交換プロトコル））のライブラリを初期化または

ロードし、メインタスクとしてプログラムモジュール80 (SILOAD, DLL) を起動することである。ジョークボックスアプリケーションのメインタスクは、ロードプログラムのライブラリを含むモジュール (SILOAD, DLL) を含み、その目的は、ライブラリ (WDLI, DLL) をロードするためにモジュールに必要な動的リンクライブラリのバージョンを確認し、それらをロードし、あるいは、ファイルを転送するためにテレコムモジュールを用いたタスクを呼び出すことである。WDLI, DLLモジュールは、動作に必要なバージョンに関するリストと、さらに、ライブラリ (LIBRARY) (DLI, DJL) またはオブジェクトファイル (DJO) によって表されるすべての機能に関するリストとをファイル (DLL, DEFAULT) に含む。それぞれのオブジェクトまたはライブラリは、オブジェクトまたはライブラリが必要とするすべての機能に関するリストと、バージョン番号および従属関係とを含む。WDLI, DLIモジュールは、すべての新しいモジュールを管理し、相互従属関係を検査して確認し、リモートロードされたモジュールが従属関係を有していないことを確認し、また、要求されたバージョンでロードされたことを確認する。ジョークボックスに固有のアプリケーション部分8は、SILOAD, DLLモジュールによってロードおよび起動されかつディスプレイの“ウィンドウズ”機能を定義するいくつかのモジュールを含み、それらには以下のものがある。すなわち、

マウスあるいはタッチスクリーン信号を処理するためのMOUSE, DJLと呼ばれるモジュール81、

オブジェクトと様々な他のモジュールとの間で交換されるメッセージを処理するためのWMESSAGE, DJLと呼ばれるモジュール82、

* ディスク上のファイルを管理するためのFIL, DLIと呼ばれるモジュール83、

ディスク上のファイルを読み出し、ディスク上にファイルを書き込むためのFIO, DJLと呼ばれるモジュール84、

タッチスクリーン、サウンドカード、および、料金処理インタフェース用ハードウェアのためのインタフェースのようなハードウェアによって生成されるすべてのイベントを監視するためのJSTRUCT, DJLと呼ばれるモジュール85。

【0043】SILOAD, DLIモジュールは、端末のタスクに固有のモジュール、すなわち、上に一覧表として記載したすべてのDLIモジュールだけでなく、ジョークボックスライブラリモジュール87のロードも管理する。ジョークボックスライブラリモジュール87は、ミキサーおよび購入のようなオブジェクトを管理するWOBJE, DJLと呼ばれるモジュール870、カーソルの移動を管理するWCURSOR, DJLと呼ばれるモジュール871、データベースを管理するDBMAPI, DJLと呼ばれるモジュール872、あらゆる種類のフォントを管理するWFO, DJLと呼ばれるモジュール873、スク립トからスクリーンを解析および生成し、かつ、“GRAMMAR, DJL”と呼ばれるモジュール876及び言語内のワードの機能を割り当てるレキシカルモジュールである“LEXY, DJL”と呼ばれるモジュール875の支援によって文法を確認するPARSER, DJL (構文解析プログラム) モジュール874からなる。PARSER, DJLモジュール874は、そのファイルヘッダーに以下の情報を含む。

* 【0044】

```
-char* parser_version_info =
    "DLL_INFO, DJL,"
    "DLL_NAME PARSE, DJL"
    "VERSION1:"
    "CREATOR KENDALF:"
    "REQUIRES lexy, djl:4"
    "REQUIRES grammar djl:5"
```

すべてのモジュールおよびすべてのライブラリ (DJO, DLL, DJL) は、PARSER, DJLモジュールの情報に類似する情報を含み、この情報は、バージョンおよび従属関係に必要なものを決定する。

【0045】したがって、PARSER, DJLモジュールは、システムによってそれが実行されるためには、LEXY, DJLモジュールバージョン4およびGRAMMARモジュールバージョン5を必要とする。様々なモジュールをお互いに接続する図2の二重矢印は、様々なファイルがロードされる順序を示す。したがって、上述したように、JUCE, EXEモジュールをロードすることから始めなければならない。それから、JUKEC

ORE, DLIモジュールをロードし、そして、JUCECORE, DLLから、GFX, DLL, TLS, DLI, WDLI, DLI, JREP, DLL, TELECOM, DLL, CRDE, DLI, および, SILOAD, DLLモジュールをロードすることができる。

【0046】一重矢印は、モジュール間の従属関係を表す。したがって、矢印91は、DJL特にDBMAPI, DJLモジュールが、CRDE, DLLに従属することを示す。矢印93は、DJOファイルがWOBJE, DJLモジュールに属することを示す。WOBJE, DJLモジュールは、それ自身がFIO, DJLモジュールに属する。矢印92aは、DB

MAPI, DJLモジュールがJSTRUCT, DJLモジュールに従属することを示し、矢印92bは、DBMAPI, DJLモジュールがWMESSEGE, DJLに従属することを示す。矢印98は、JSTRUCT, DJLモジュールがWMESSEGE, DJLモジュールに従属することを示す。WMESSEGEモジュールは、MOUSE, DJLモジュールに従属し、また、FILIO, DJLモジュールは、FIL, DJLモジュールに従属するので、モジュールXCP, DJPは、矢印856で示されるように、JSTRUCT, DJLモジュールに従属し、かつ、矢印826で示されるように、WMESSEGE, DJLモジュールに従属する。JHANDLERモジュールは、矢印97で示されるように、WMESSEGE, DJLモジュールに従属し、また、矢印96で示されるように、JSTRUCT, DJLモジュールに従属する。SILLOP, DJLモジュールは、矢印95で示されるように、JSTRUCT, DJLモジュールに従属し、また、矢印94で示されるように、WMESSEGE, DJLモジュールに従属する。TELECOM, DLLモジュールは、矢印910で示されるように、JEEP, DLLモジュールに従属し、そして、JEEP, DLLモジュールは、矢印759で示されるように、WDL, DLLモジュールに従属する。WDL, DLLモジュールは、矢印725で示されるように、TLS, DLLモジュールに従属する。同様に、矢印99cは、GRAMMAR, DJLモジュールがLEXY, DJLモジュールに従属することを示し、また、矢印99bで示されるように、LEXY, DJLモジュールは、PARSER, DJLモジュールに従属する。したがって、上述したことからわかるように、PARSER, DJLモジュールは、それ自身を実行するためには、LEXY, DJLおよびGRAMMAR, DJLを必要とし、バージョン1のPARSER, DJLモジュールは、バージョン4のLEXY, DJLモジュールおよびバージョン5のGRAMMAR, DJLモジュールを呼び出す。同様に、WOBJECT, DJLモジュールは、矢印99aで示されるように、PARSER, DJLモジュールに従属する。したがって、すべての、DJO, DLL, および、DJLモジュールおよびライブラリは、PARSERモジュールの情報に類似する情報を含み、それは、与えられたモジュールが従属する様々なモジュールのバージョン要件を決定する。また、この情報は、図2の矢印で示されるように、モジュールの他のモジュールに対する従属関係を示す。

【0047】さらに、ライブラリロードモジュールSILLOAD, DLLは、モジュールSILLOP, DLL90をイベント待ちテープでもってロードまたは起動する。一組のモジュール88は、ジョークボックス型端末のモニター上に提供されるウィンドウズディスプレイの

ウィンドウを管理するためにそこに含まれなければならないファイルのリストを含む。

【0048】このオブジェクトのリストは以下のものから構成される。すなわち、

… モニター上のメインページを規定するのを可能にするオブジェクトファイル“WPSCREEN, DJO”883、

… このメインページで利用可能なスクリーンの数を規定するのを可能にし、それによって、複数のウィンドウまたはスクリーンを表示するのを可能にするオブジェクトファイル“WSCREEN, DJO”881、

… スクリーンが使用する画像をスクリーン上に決定および規定するのを可能にするオブジェクトファイル“WIMAGE, DJO”880、

… 画像が動画化される場合の動画を規定するのを可能にするオブジェクトファイル“WANIM, DJO”882、

特許出願PCT WO96/12258号に記載されるグラフィックインタフェースに使用される操作ボタンのようなメインページのスクリーン上で使用されるボタンを規定および管理するのを可能にするオブジェクトファイル“WBUTTON, DJO”885、

… オブジェクトの先頭に書き込むことを可能にするラベルを生成するのを可能にするオブジェクトファイル“WLABEL, DJO”884、

… 仮想スクロール表示領域を規定するのを可能にするオブジェクトファイル“WSCROLLER, DJO”886。

【0049】“JHANDLER”と呼ばれるライブラリは、スクリーンの所定の使用法を規定するのを可能にし、したがって、上述のモジュールによって規定された様々なオブジェクトへのリンクを保証するのはどのインタフェースかを決定するのを可能にする。

【0050】ライブラリモジュール“XCP, DJP”88は、チケット受領システムあるいは硬貨またはカードによる支払いユニットの支払いタスクのような支払いタスクを管理するのを可能にし、また、基本情報を、ユーザの暗証コードを記憶するための集積回路であるIBUTTONにバックアップするのを可能にする。

【0051】したがって、リモートロードによって新しいファイルがシステムに送られたとき、ファイルは、そのレベルに関する情報を含み、それは、ファイルの種類に依存する。例えばWIMAGE, DJOのようなグラフィック画像のファイルは、もっとも高いレベルを有し、例えばXCP, DJPのようなハードウェア管理ファイルは、もっとも低いレベルを有する。JEEP, DLLモジュールは、もっとも低いレベルのファイルから開始してより高いレベルのファイルに進むことによって従属論理を確認し、その間ずっと、ファイルまたはライブラリ間の必要な従属関係が守られていることを保証す

る。このように、WOBJECT、DLLモジュール内の変更によって、JEEP、DLLモジュールは、従属しかつそれを実行するために必要なDLLファイルごとに、WOBJECT、DLLモジュールに必要とされるバージョン情報が、WOBJECT、DLLモジュールに記録された情報に必要とされる最低限のバージョンに対応するかを確認することになる。したがって、もしWOBJECT、DLLモジュールがWPSCREEN、DJOMジュールのある特定のバージョンを必要とすれば、JEEP、DLLモジュールは、そのバージョンが少なくとも存在することを確認し、しかも、不十分なレベルのバージョンしか存在しない場合には、それは、問題を報告する。そして、JEEP、DLLモジュールは、FILLIO、DLLモジュールおよびFIL、DLLモジュールへの従属リンクに進む。

【0052】ジュークボックスのハードディスクは、ディレクトリC:\NEWJUKEを備えるように構成され、そのディレクトリは、新しいモジュールがリモートロードされたときの新しいジュークボックスファイルを含む。もう1つのファイルC:\OLDJUKEは、ジュークボックスが動作するのに必要な安定したバージョンのファイルおよびモジュールからなるバックアップを含む。JEEP、DLL(JUKE EXECUTION EXCHANGE PROTOCOL)モジュールは、ある一定の期間だけ古いファイルをバックアップすることによって、また、それらのファイルをNEWJUKEディレクトリに移すことによって更新されるモジュールおよびファイルの記録を取る自動ファイルマネージャを含む。さらに、このタスクは、リモートロード動作中に突然にイベントが発生するとファイルをディスクのトラックにコピーする。また、JEEP、DLLモジュールは、いったん自動ファイルマネージャがジュークボックスの更新がなされたと判断してしまえば、ジュークボックスのファイルの実行レベルの変更に関する責任を有するリブートマネージャを含む。また、JEEP、DLLモジュールは、従属関

係が失われていることを検出すると、MISDEPS、DATファイルを生成する。このファイルは、「NEEDPARSER、DLL 矢印 バージョン21、すなわち1NEEDLEXY、DLL version2、0」などの形態の行を含む。このファイルによって、サーバは、このMISDEPS、DATファイルを読み込むことによって、ジュークボックス上のモジュールを判定することができ、かつ、それらを再ロードすることができる。

【0053】当業者は、本発明の範囲内においてその他の変形を考えることができる。

【図面の簡単な説明】

【図1】本発明を構成するハードウェアの電氣的な構成図である。

【図2】ライブラリモジュールおよびオブジェクトモジュール間の関係を示すフローチャートである。

【符号の説明】

- 2 大容量記憶装置制御回路
- 3 入力制御回路
- 4 通信制御回路
- 5 音声制御回路
- 6 ディスプレイ制御回路
- 21 大容量記憶装置
- 26、36、46、56、57、66、67 バッファ
- 31 遠隔制御アセンブリ
- 32 キーロック
- 33 タッチスクリーン
- 34 キーボード
- 35 料金支払い装置
- 41 モデム
- 53 同調増幅器
- 54 スピーカ
- 55 マイク
- 61 テレビ
- 62 モニタ

【図1】

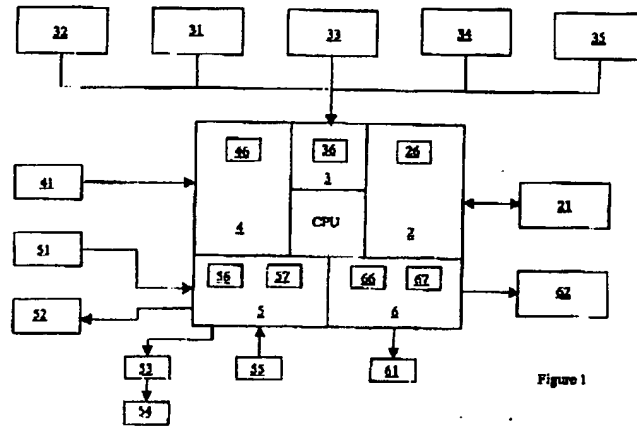
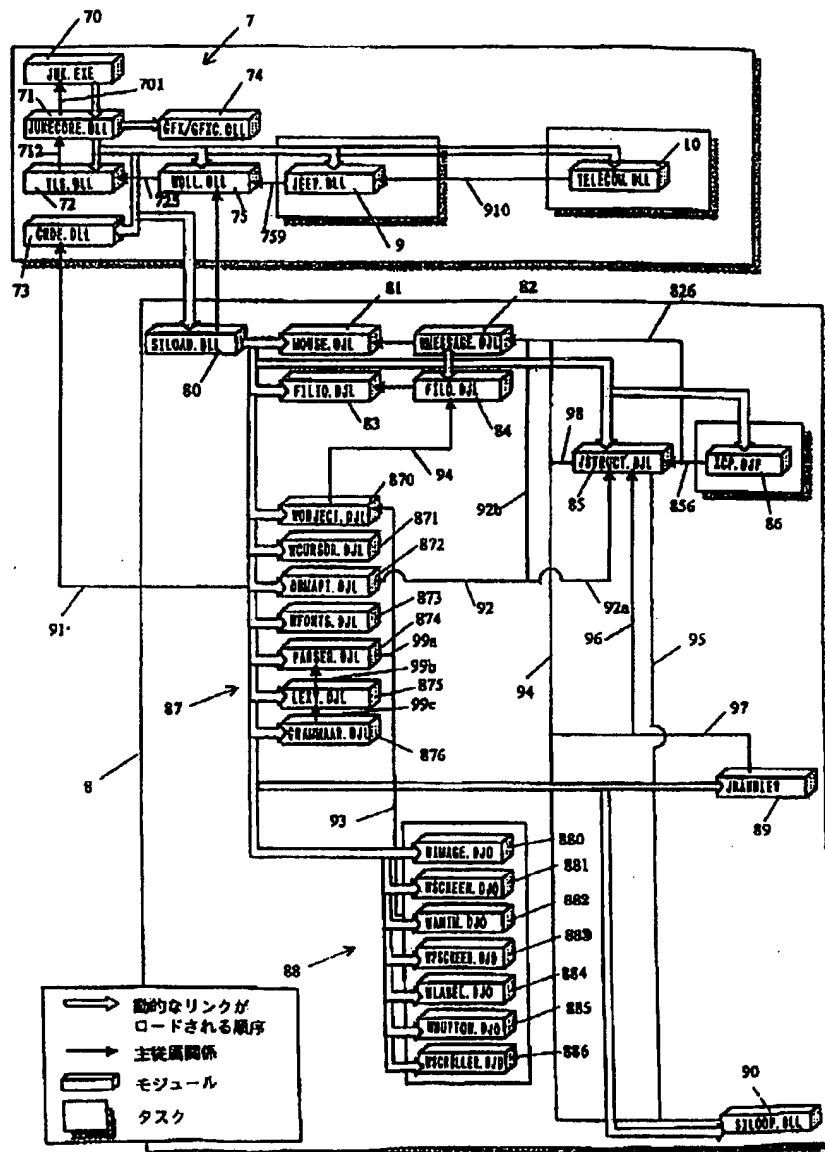


Figure 1

【図2】

Figure 2



フロントページの続き

(72)発明者 ドミニク・デイオン
 カナダ国、ケベック、ラバル・アツシエ・
 7・エル・4・エン・8、イボン・ヘルシ
 エ・411

【外語明細書】

1. Title of Invention

**System for remote loading of objects or files in order
to update software**

2. Claims

1. System for remote loading of objects or files in order to update software, wherein the operating system architecture includes a breakdown of various tasks into software modules that are interconnected by dynamic links or are made up of executable subroutines that have primary dependence links to other parts of the operating system, wherein each of the software modules is composed of object files or libraries that are represented by dynamic link libraries which are organized between them according to a number of levels of dependence that are described in their respective attributes.

2. System for remote loading of objects or files in order to update software according to claim 1, wherein the attributes of an object or a library indicate a version number and the dependencies of the object with respect to other objects.

3. System for remote loading of objects or files in order to update software according to claim 2, wherein the attributes indicate the levels attributed to the software modules.

4. System for remote loading of objects or files in order to update software according to claims 1 to 3, wherein the various tasks include a main task, which includes a module (JUKECORE.DLL), which is intended to load the dynamic link libraries (DLL), to initialize a kernel of the program, to initialize the graphics management module (GFX.DLL), to initialize a library loading module (WDLL.DLL), to load the Telecom module of the telecommunications tasks (TELECOM.DLL), and to launch the screen interpreter program as a main task.

5. System for remote loading of objects or files in order to update software according to claim 4, wherein the program is subdivided into a certain number of modules, each of which defines a task that is specific to a terminal that is equipped with at least a telecommunication modem and a central unit.

6. System for remote loading of objects or files in order to update

software according to claim 5, wherein said specific task is the task that corresponds to a jukebox and includes a specific main task, along with a certain number of modules which define the window functions of the display, which are as follows:

- a MOUSE.DJL module for exploiting mouse or touch-screen signals;
- a WMESSAGE.DJL module for exploiting messages that are exchanged among the objects;
- a FIL.DJL module for managing files on disk;
- a FILIO.DJL read-write module for files on disk;
- a JSTRUCT.DJL module for monitoring events generated by hardware, such as the touch screen, a sound card, and a processing interface for coin equipment.

7. System for remote loading of objects or files in order to update software according to claim 5 or 6, wherein the specific main task of the jukebox application includes a module (SILOAD.DLL) that contains the library of the loader program that is designed to verify the required dynamic link library versions, to load them, and to call the Telecom task module in order to effect file transfer.

8. System for remote loading of objects or files in order to update software according to claim 7, wherein said SILOAD.DLL module includes in a file (DLL.DEFAULT) a list of minimum versions that are required for operation, as well as a list of all functions that are represented either by libraries (DATA JUKEBOX LIBRARY) or by object files (DJO, DATA JUKEBOX OBJECT).

9. System for remote loading of objects or files in order to update software according to claim 8, wherein each object or library contains the list of all of the functions that the library or the object needs, as well as the version numbers and dependencies.

10. System for remote loading of objects or files in order to update software according to claim 4, wherein WDLL.DLL module includes means for

managing new modules and verifying that remotely loaded modules have no missing dependencies and that they have been loaded with the necessary versions.

11. System for remote loading of objects or files in order to update software according to claim 7, wherein the SILOAD.DLL module includes means for managing loading of the modules that are specific to the task of the terminal, including all DJL modules and jukebox library modules constituted by WOBJECT.DJL module which manages objects, a mixer, and purchases; the WCURSOR.DSL module, which manages movements of the cursor; a DBMAPI.DJL module which manages the database; a WFONT module which manages the font types; the PARSER.DJL module which analyzes and generates the screens based on the script and verifies grammar with the aid of the "GRAMMAR.DJL" module and the "LEXY.DJL" lexical module.

12. System for remote loading of objects or files in order to update software according to claim 11, wherein the SILOAD.DLL library loading module includes a "WINDEF" module, which contains a list of the files that have to be included in order to manage the windows of a display that is shown on a monitor of the jukebox type terminal.

13. System for remote loading of objects or files in order to update software according to claim 12, wherein said objects list is composed of :

- an "OBJET WPSCREEN.DJO" object module which makes it possible to define a main page on the monitor;

- a "WSCREEN.DJO" module, which makes it possible to define in said main page the number of available screens and thus to make it possible to display multiple windows or screens;

- a "WIMAGE.DJO" module, which makes it possible to determine and define on the screen the image that it will use;

- a "WANIM.DJO" module, which makes it possible to define animation when the image is animated;

- a "WBUTTON.DJO" module, which makes it possible to define and manage buttons that are used on the screen of the main page;

- a "WLABEL.DJO" module, which makes it possible to create labels that make it possible to write on an object; and

- a "WSCROLLER.DJO" module, which makes it possible to define display zones that are to be scrolled between two points.

14. System for remote loading of objects or files in order to update software according to claim 13, wherein all of these object modules that are managed by the main task use a "JHANDLER" library, which makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the links to the different objects defined by the previous software modules.

15. System for remote loading of objects or files in order to update software according to claim 7, wherein the SILOAD.DLL task includes means for launching or loading an "XCP.DJL" module, which makes it possible to manage payment tasks such as ticket receiving systems or coin or card payment units and also makes it possible to protect basic information in the IBUTTON.

3. Detailed Description of Invention

This invention pertains to a system for remote loading of objects or files in order to update software, particularly for audiovisual reproduction systems that are triggered by the payment of fees, such as jukeboxes or other devices.

In the prior art, devices for remote loading of an operating system through a network are known such as, for example, from British Patent No. 2 231 180. The teaching of this patent application calls for loading a first computer, which is connected to a second computer via a network, by loading a subset of the operating system into the memory of the first computer, whereby said subset contains the commands that make it possible to copy files, create a directory, and format a disk, as well as allowing a connection to operate through the network so that this subset can then be used to transfer all of the operating system files from the second computer to the disk of the first computer.

In this type of remote loading, the purpose is to load the entire operating system and all of the operating system files. This thus limits remote loading either to tying up, for relatively long periods of time, telecommunications systems that are to carry out remote loading in the case of the operating system, or causes the relevant files to have to be updated frequently.

From U.S. Patent No. 4,958,278 a system is also known for remote loading to a terminal that is not equipped with a disk player.

From French Patent No. 2 682 786 another process is known for remote loading to a terminal that is not equipped with a hard disk.

Finally, European Patent No. 0 358 992 teaches a method for remote loading, via a network, of an operating system or of executable files to a computer that does not include a boot device or other devices that hold the executable program. A first minimum boot program is transferred initially, and then this minimum boot program executes itself, establishes a logical link to a

disk of the server, and allows the querying computer to treat the server disk as a local boot device.

The object of the invention is to avoid the necessity, on the one hand, of rebooting the terminal to which downloading is done and, on the other, to make it possible to transfer operating files or parts of an executable program without having to reinitialize the machine, and doing so while making sure that the operation of the system is not degraded by the remotely loaded version.

This object is achieved by virtue of the fact that the architecture of the operating system provides for breaking the different tasks down into software modules that are interconnected by means of dynamic links or are composed of executable subroutines that have main dependence links to other parts of the operating system, whereby each of the modules is composed of object files or libraries that are represented by dynamic link libraries, which are organized among themselves according to a number of dependence levels, as described in their respective attributes.

According to another feature, the attributes of an object or a library indicate the version number and the dependencies of the object with respect to other objects.

According to another feature, the attributes indicate the levels attributed to the modules.

According to another feature, the different tasks include a main task that includes a module (JUKECORE), which is designed to load the dynamic link libraries (DLLs), to initialize the nucleus of the program, to initialize the graphics management module (GFX), to initialize the library loading module (WDLL), to load the Telecom module of the telecommunications tasks (TELECOM.DLL), and to launch the screen interpreter program as a main task.

According to another feature, the program is subdivided into a certain number of modules that define a task that is specific to the terminal.

According to another feature, this specific task is that which corresponds

to a jukebox and includes a specific main task, a certain number of modules that define the "Windows" functions of the display, which are the following:

- a module for running the mouse signals or the touch screen;
- a module for running the messages that are exchanged among the

objects:

- a "FIL.DJL" module for managing the files on disk;
- a "FIL.DJL" module for reading and writing files to and from disk;
- a "FILIO.DJL" module for monitoring all of the events that are generated

by the hardware, such as the touch screen, the sound card, and the money hardware processing interface.

According to another feature, the main task of the jukebox application includes a "SILOAD.DLL" module, which contains the library of the loader program, whereby the library is intended to verify the versions of the dynamic link libraries that are requested, to load them, and to call the Telecom task modules in order to transfer files.

According to another feature, the SILOAD module includes the list in a file (DLL.DEFAULT) of the minimum versions that are required for operation, as well as the list of all of the functions that are represented either by the libraries (DJL) (DATA JUKEBOX LIBRARY) or by the object files (DJO DATA JUKEBOX OBJECT).

According to another feature, the object or library contains the list of all of the functions that the library or object needs, as well as the version numbers and dependencies.

According to another feature, WDLL ensures the management of all of the new modules and verifies that the remotely loaded modules do not have any missing dependencies and that they have been loaded with the necessary versions.

According to another feature, SILOAD manages the loading of the modules that are specific to the task of the terminal, i.e., all of the "DJL"

modules already listed above, as well as that the jukebox library modules constituted by WOBJECT manage the object, the mixer, and the purchases; the "WCURSOR" module manages the movements of the cursor; the DBMAP module manages the database; the "WFONTs" module manages all of the font types; the "PARSER" module analyzes and generates the screens from the script and verifies the grammar with the aid of the "GRAMMAR.DJL" module and the lexical module "LEXY.DJL."

According to another feature, the library loading module SILOAD includes a "WINDEF" module that contains the list of the files that have to be included in order to manage the windows of a Windows display that is supplied on the monitor of the jukebox-type terminal.

According to another feature, this list of objects consists of:

- an "OBJET WPSCREEN.DJO" module, which makes it possible to define the main page on the monitor;
- a "WSCREEN.DJO" object module, which makes it possible to determine in this main page the number of screens that are available and thus to allow multiple windows or screens to be displayed;
- a "WIMAGE.DJO" module, which makes it possible to determine and define on the screen the image that it will use;
- a "WANIM.DJO" module, which makes it possible to define the animation when the image is animated;
- a "WBUTTON.DJO" module, which makes it possible to define and manage the buttons that are used on the screen of the main page;
- a "WLABEL.DJO" module, which makes it possible to create the labels that make it possible to write on top of an object; and
- a "WSCROLLER.DJO" module, which makes it possible to design the scroll display zones, between two points for example, horizontal, diagonal, vertical.

According to another feature, all of these object modules, which are

managed by the main task, use a "JHANDLER" library, which makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the link to the different objects that are defined by the preceding modules.

According to another feature, the SILOAD task launches or loads the "XCP" module, makes it possible to manage payment tasks such as those handled by ticket receiving systems or coin or card payment units, and also makes it possible to save the basic information in the IBUTTON.

Other features and advantages of this invention will be made clearer by reading the description given below, with reference to the attached drawings.

Preferably, but without being limited hereto, the audiovisual reproduction system uses the hardware elements listed and referenced below.

Referring to figure 1, a microprocessor central unit 1 is a high-performance PC compatible system, whereby during implementation a Pentium-type system was chosen that has the following memory resources and specifications:

- compatibility with the local VESA bus;
- processor cache: 256 KB minimum;
- RAM memory: 32 MB;
- high-performance serial ports;
- SVGA microprocessor graphics adapter;
- SCSI/2 type bus controller;
- self-powered static RAM memory.

Any other central processing unit that has the equivalent or better

performance can be used in the invention.

This central processing unit controls and manages a sound control circuit (5), a telecommunications control circuit (4), an input control circuit (3), a bulk storage control circuit (2), and a display control circuit (6). The display consists primarily of a flat-screen non-interlaced SVGA high-resolution and low-radiation video monitor (82), i.e., this is the monitor that is used to reproduce images (for example, the album covers of musical selections), graphics, or video clips.

A bulk storage (21), which uses hard disks of the high-speed and high-capacity SCSI type, is associated with the storage that is already present in the microprocessor device. These storage serves to store digitized and compressed audiovisual information.

A high-speed telecommunications modem adapter (41), is built in in order to enable the link to an audiovisual information distribution network that is controlled by a central server.

In order to reproduce the sound information of the musical selections, the system includes loudspeakers (54) that receive the signal from a tuner-amplifier (53), which is connected to electronic circuit (5) that incorporates two memory buffers (56, 57) of the music synthesizer type that are supplied to support a large number of input sources while providing CD (compact disc) type output quality.

Likewise, the display control circuit (6) also has two buffer memories (66, 67) for the purpose explained above.

A 240-watt thermally regulated and ventilated power supply feeds power to the system. This power supply is protected against overcurrents and overdriving.

By means of its input controller circuit (3), the audiovisual reproduction system manages a touch screen (33), which includes a glass coating panel that uses the "advanced surface-wave technology", as well as an AT-type bus controller. This touch screen is able to display on video monitor (82) or the

screen of a television set (61) various selection data that are used by the customers, as well as management command and control data that are used by the manager or owner of the system. This touch screen is also used for maintenance purposes in combination with an external keyboard (34) that can be connected to the system which, for this purpose, has a keyboard connector that is controlled by a key lock (32) via interface circuit (3).

Input circuit (3) also interfaces the system with a remote control assembly (31) that consists of, for example, a radio frequency RF remote control.

A fee payment device (35) is also connected to input interface circuit (3). It is also possible to use any other device that makes it possible to receive any kind of payment using coins, tickets, tokens, magnetic or microchip cards, or any combination of these means of payment.

To allow the system to be installed, it is equipped with a chassis or is built with customizable external fittings.

In addition to these elements, a wireless microphone (55) is connected to sound controller (5); this makes it possible to convert the sound controller into a powerful public-address and information system or optionally into a karaoke machine. Likewise, the system can also use a wireless loudspeaker system.

Remote control assembly (31) allows the manager from, e.g., behind the bar, to access and control various commands such as:

- the microphone on-off control;
- the loudspeaker muting control;
- the control for canceling the musical selection that is currently being

played.

Two buffers (56, 57) are associated with sound controller circuit (5) in order to make it possible to store, each alternately, a data item corresponding to at least approximately a quarter of a second of sound. Likewise, two buffers (66, 67) are associated with video controller circuit (6), whereby each buffer is able, either by itself or alternatively, to store at least approximately one-tenth of

a second of an image. Finally, a respective buffer (46, 36, 26) is associated with each of communications controller circuit (4), input interface circuit (3), and storage circuit (2).

The system operating software was developed around a library of tools and services that were very largely oriented toward the audiovisual domain in a multimedia universe. This library advantageously includes a high-performance multitask operating system that effectively allows the simultaneous execution of multiple fragments of code. This operating software also allows the concurrent execution, in an orderly and completely conflict-free way, of operations that are carried out on the display and the sound reproduction structure, as well as the managing of the telecommunications links through the distribution network. Moreover, this software is highly flexible.

The operating system is divided into modules, which include a first boot module (7), which in turn is subdivided into a first main program module (70) called "JUK.EXE", which checks the memory and verifies whether the minimum number of objects is available to ensure the operation of the jukebox; a second module (71), which is dynamically linked and is dependent thereon, is represented by the module called "JUKECORE.DLL". The function of the second module (71) is to hold the libraries in C and to ensure the execution of the main task.

The architecture of the operating system calls for the different tasks to be broken down into software modules that are connected to one another by dynamic links or consist of executable subroutines that have main dependence links to other parts of the operating system. Each of the modules is composed of object files or dynamic link library files that are organized according to a number of dependence levels described in the attributes. The attributes of an object or a library indicate the version number and the dependencies of the object or library file with respect to other object files, as described below for the module called "PARSER". Each attribute indicates the level attribute to the

module. Thus, the JUK.EXE module (70) is of a higher level than the modules JUKECORE.DLL (71), TLS.DLL (72), CRDE.DLL (73), GFX.DLL (74), WDLL.DLL (75), JEEP.DLL (9) and TELECOM.DLL (10), but TELECOM.DLL module (10) is dependent on JEEP.DLL module (9) (see link 910), and is thus lower in level than JEEP.DLL module (9).

Likewise, JEEP.DLL module (9) (see link 759) is of a lower level than WDLL module (75), because it is dependent on it and TLS.DLL module (725), of a higher level than WDLL.DLL module (75). However, TLS.DLL and GFX.DLL modules may be on the same level. The main task includes a module (JUKECORE.DLL) whose purpose is to initialize or load module (73), the nucleus of program called "CRDE.DLL", to initialize or load module (74) for managing graphics (GFX.DLL), to initialize or load module (75) for loading libraries (WDLL.DLL), to load Telecom module (10) for the task of telecommunications (DLL), to load TLS.DLL module (72), which contains all the uses required for the jukebox, for telecommunications, time, decryption, etc.,..., to initialize or load the library of JEEP.DLL (Juke Execution Exchange Protocol) programs, which handle the tasks of an integrity server, load request and dialogue with the server, and to launch the module of program (80, SILOAD.DLL) as a main task. The main task of the jukebox application includes a module (SILOAD.DLL) that contains the library of the loader program whose purpose is to verify the dynamic link library versions required in the module for loading libraries (WDLL.DLL), to load them or to call the tasks using the Telecom module in order to transfer files. The WDLL.DLL module includes the list in a file (DLL.DEFAULT) of the versions that are required for operation, as well as the list of all of the functions that are represented by libraries (LIBRARY) (DLL, DJL) or by object files (DJO). Each object or library contains the list of all of the functions that the library or object needs, as well as the version members and dependencies. The WDLL.DLL module manages all of the new modules, checks to verify the interdependencies, and verifies that the remotely loaded

modules have no dependence and have been loaded with the required versions. Application part (8) that is inherent in a jukebox includes a certain number of modules that are loaded and launched by SILOAD.DLL module and that define the "Windows" functions of the display, which include the following:

- a module (81) called MOUSE.DJL for running the mouse or touch-screen signals;
- a module (82) called WMESSAGE.DJL for running the messages that are exchanged between the objects and the various other modules;
- a module (83) called FIL.DLL for managing the files on disk;
- a read-write module (84) called FIL O.DJL for files on disk;
- a module (85) called JSTRUCT.DJL for monitoring all of the events produced by the hardware, such as the touch screen, the sound card, and the interface for processing fee hardware.

SILOAD.DLL module manages the loading of the modules that are specific to the task of the terminal, i.e., all of the DJL modules already listed above, as well as jukebox library modules (87), consisting of a module called WOBJECT.DJL (870), which manages the objects such as the mixer and purchases; a module (871) called WCURSOR.DJL, which manages the movements of the cursor; module (872) called DBMAPI.DJL, which manages the database; a module (873) called WFONT.S.DJL, which manages all of the types of fonts; PARSE.R DJL (syntactic analysis program) module (874), which analyzes and generates the screens from the script and verifies grammar with the aid of a module called "GRAMMAR.DJL" (876) and a module called "LEXY.DJL" (875), which is the lexical module that assigns the functions of the words within the language. PARSE.R.DJL module (874) contains in its file header the following information:

```
-char*parser _version _info =  "DLL_INFO.DJL,"
                                "DLL-NAME PARSE.R DJL"
                                "VERSION 1;"
```

"CREATOR KENDALF;"

"REQUIRES lexyy.djl;4"

"REQUIRES grammar-djl;5";

All of the modules and all of the libraries (DJO, DLL, DJL) contain information similar to that of the PARSER.DJL module, and this information determines the needs in terms of versions and dependence.

Thus, the PARSER.DJL module needs LEXY.DJL module version 4 and GRAMMAR module version 5 in order to allow it to be executed by the system. The double arrows in Figure 2, which connect the various modules to one another, indicate the order in which the different files are loaded. Thus, as indicated previously, it will be necessary to start by loading JUKE.EXE module, then loading JUKECORE.DLL module, and being able, from JUKECORE.DLL, to load GFX.DLL, TLS.DLL, WDLL.DLL, JEEP.DLL, TELECOM.DLL, CRDE.DLL, and SILOAD.DLL modules.

The single arrows represent the dependencies between modules. Thus, arrow (91) indicates that the DJL and, in particular, DBMAPI.DJL modules are dependent on CRDE.DLL. Arrow (93) indicates that the DJO files are dependent on the WOBJECT.DJL module. The WOBJECT.DJL module is itself dependent on the FILIO.DJL module. Arrow (92a) indicates that DBMAPI.DJL module is dependent on JSTRUCT.DJL module, and arrow (92b) indicates that DBMAPI.DJL module is dependent on WMESSAGE.DJL. Arrow (98) indicates that JSTRUCT.DJL module is dependent on the WMESSAGE.DJL module. WMESSAGE module is dependent on the MOUSE.DJL module and, since FILIO.DJL module is dependent on the FIL.DJL module, the module XCP.DJP is dependent, as indicated by arrow (856), on JSTRUCT.DJL module and, as indicated by arrow (826), on WMESSAGE.DJL module. The JHANDLER module is dependent, as indicated by arrow (97), on WMESSAGE.DJL module and, as shown by arrow (96), on JSTRUCT.DJL module. The SILOOP.DJL module is dependent, as indicated by arrow (95), on JSTRUCT.DJL module

and, as indicated by arrow (94), on WMESSAGE.DJL module. The TELECOM.DLL module is dependent, as indicated by arrow (910), on JEEP.DLL module, which in turn is dependent, as shown by arrow (959), on WDLL.DLL module. The WDLL.DLL module is dependent, as indicated by arrow (725), on TLS.DLL module. Likewise, arrow (89c) shows that GRAMMAR.DJL module is dependent on LEXY.DSL module and, as indicated by arrow (99b), that LEXY.DJL module is dependent on PARSER.DJL module. Thus, as was seen previously, PARSER.DJL module requires LEXY.DJL and GRAMMAR to execute itself, and version 1 of PARSER.DJL module calls version 4 of LEXY.DJL module and version 5 of GRAMMAR.DJL module. Likewise, WOBJECT.DJL module is dependent, as indicated by arrow (99a), on PARSER.DJL module. Thus, all of the .DJO, .DLL, and .DJL modules and libraries contain information similar to that of the PARSER module, which determines the version requirements of the various modules on which a given module is dependent. This information also indicates the dependencies of the modules with respect to the other modules, as indicated by the arrows in Figure 2.

Library loading module SILOAD.DLL also loads or launches a module SILOOP.DLL (90), with an event wait tape. A set of modules (88) contains a list of the files that have to be included there to manage the windows of a Windows display that is provided on the monitor of the jukebox-type terminal.

This list of objects consist of:

- an object file (883) "WPSCREEN.DJO", which makes it possible to define the main page on the monitor;
- an object file (881) "WSCREEN", which makes it possible to determine on this main page the number of screens available and thus to make it possible to display multiple windows or screens;
- an object file (880) "WIMAGE.DJO", which makes it possible to determine and define on the screen the image that the screen will use;

- an object file (882) "WANIM.DJO", which makes it possible to define the animation when the image is animated;

- an object file (885) "WBUTTON.DJO", which makes it possible to define and manage the buttons that will be used on the screen of the main page, such as the actuation buttons used in the graphical interface that is defined in patent application PCT WO 96/12258;

- an object file (884) "WLABEL.DJO", which makes it possible to create labels that make it possible to write on top of an object; and

- an object file (886) "WSCROLLER.DJO", which makes it possible to define the vertical-scroll display zones.

A library called "JHANDLER" makes it possible to define the fixed uses of the screens and thus to determine which are the interfaces that ensure the links to the different objects that are defined by the previous modules.

Library module "XCP.DJP" (86) makes it possible to manage the payment tasks such as those of ticket receiving systems or coin or card payment units and also makes it possible to back up basic information in the IBUTTON, which is an integrated circuit for storing secret codes for the user.

Thus, when a new file is sent by remote loading to the system, the file contains information on its level, which depends on the type of file. The files of the graphical images, for example WIMAGE.DJO, have the highest levels, and the hardware management files, for example XCP.DJP, have the lowest levels. The JEEP.DLL module verifies the dependence logic by starting with the lowest-level files and moving up toward the higher files, all the while ensuring that the necessary dependencies between the files or libraries are respected. In this way, a modification in WOBJECT.DJO module will cause JEEP.DLL module to verify that the version information required by WOBJECT.DJO module for the DJO files that are dependent and are required for its execution corresponds to the minimum versions required by the information recorded in the WOBJECT.DJO module. Thus, if WOBJECT.DJO module requires a certain

version of WPSCREEN.DJO module, it will verify that this version is at least present and that, if there is only a version of an inadequate level, it will report a problem. Then JEEP.DLL module will go up the dependence links toward FILIO.DJL and FIL.DJL modules.

The hard disk of the jukebox is organized in such a way as to comprise a directory C:\NEWJUKE, which contains the new jukebox files when new modules are remotely loaded. Another file, C:\OLDJUKE, contains a backup of the stable versions of the files and modules that are required for the operation of the jukebox. The JEEP.dll (JUKE EXECUTION EXCHANGE PROTOCOL) module contains an automatic file manager that keeps track of the modules and files that are updated by backing up the old files for a certain period of time and by moving their files into the NEWJUKE directory. This task also copies the files on the tracks of the disk in the event that there is a sudden incident during the remote loading operation. The JEEP.OLL module also contains a reboot manager that is responsible for changing the execution level of the files of the jukebox once the automatic file manager has determined that an update of the jukebox has been accomplished. The JEEP.DLL module also generates a MISDEPS.DAT file when a missing dependence has been detected. This file contains lines in the form NEEDPARSER.DLL arrow version 2: NEEDLEXY.DLL -> version 2.0, etc.... This file allows the server, by reading this MISDEPS.DAT file, to determine the modules on the jukebox and to reload them.

Other modifications within the grasp of one skilled in the art are also part of the spirit of the invention.

4. Brief Description of Drawings

Figure 1 shows a schematic of the electrical diagrams of the hardware that constitutes the invention.

Figure 2 shows a flow chart of the relationships between the library modules and the object modules.

Fig. 1

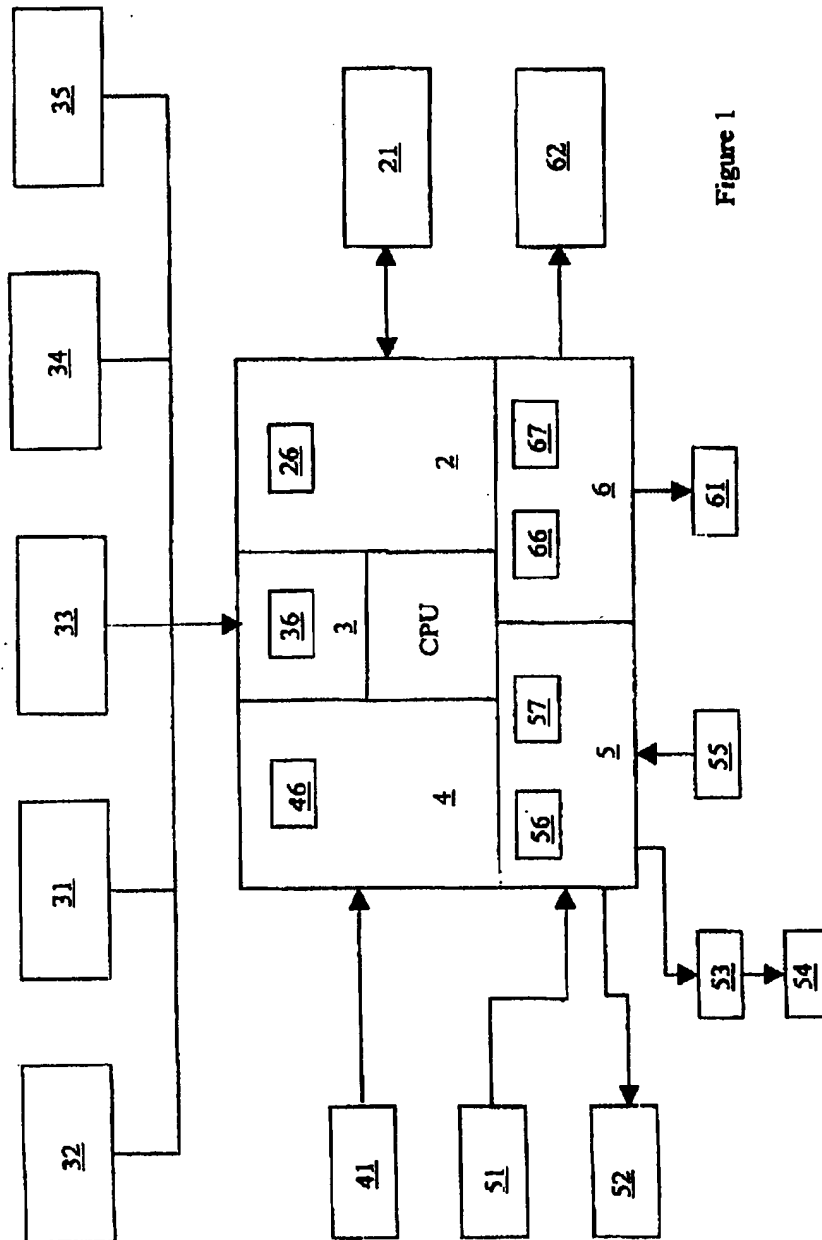
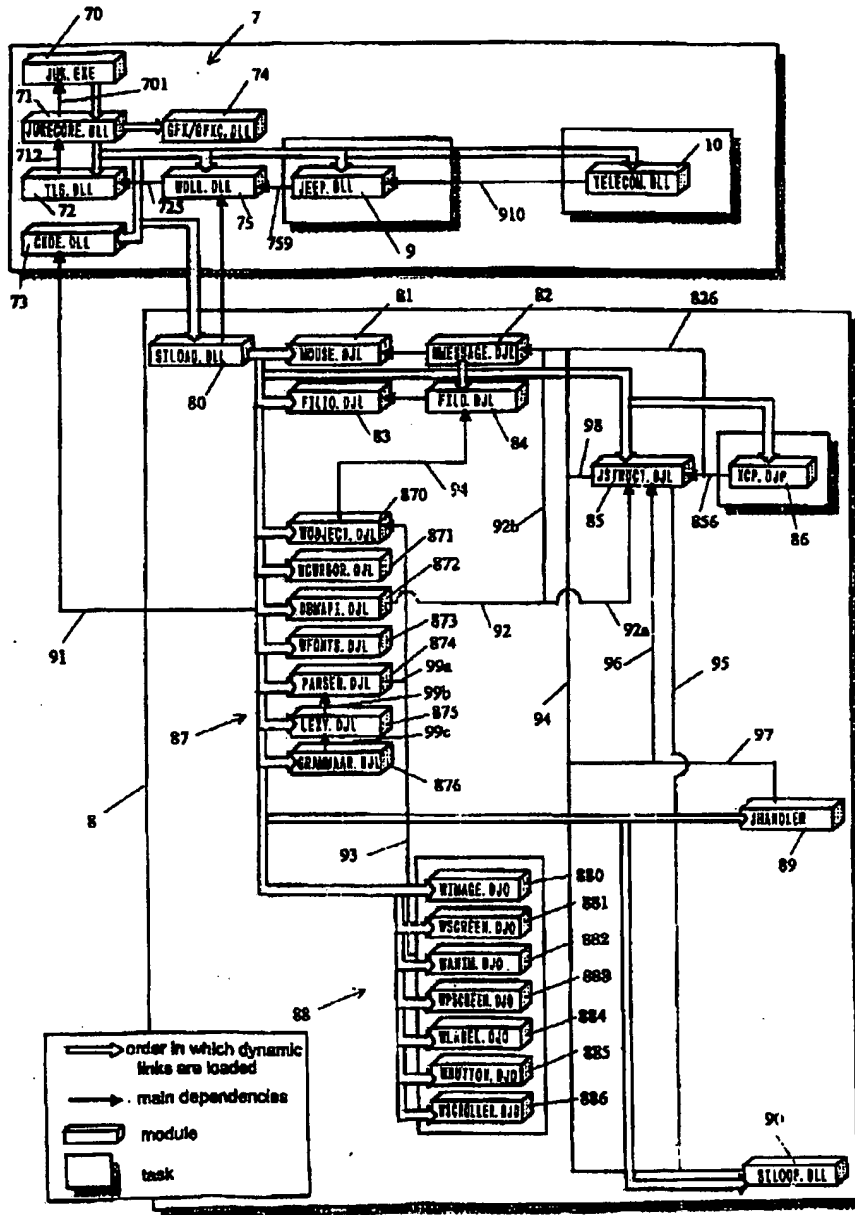


Figure 1

Fig. 2

Figure 2



1. Abstract

This invention pertains to a system for remote loading of objects or files in order to update software.

System for remote loading of objects or files in order to update software, wherein the operating system architecture includes a breakdown of various tasks into software modules that are interconnected by dynamic links or are made up of executable subroutines that have primary dependence links to other parts of the operating system, wherein each of the software modules is composed of object files or libraries that are represented by dynamic link libraries which are organized between them according to a number of levels of dependence that are described in their respective attributes.

2. Representative Drawing

Fig. 1

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.